A11100 994631

NBS PUBLICATIONS

NBSIR 81-2328

A Manual for Designing and Implementing A Process to Monitor Complex System Developments

September 1981

Prepared by

Experimental Technology Incentives Program
National Bureau of Standards
Washington, DC 20234

QC
100
.U56
81-2328
1981
c.2

THE EXPERIMENTAL TECHNOLOGY INCENTIVES PROGRAM

The Experimental Technology Incentives Program (ETIP) of the National Bureau of Standards pursues an understanding of the relationships between government policies and technology-based economic growth. The pursuit of this objective is based on three premises:

- Technological change is a significant contributor to social and economic development in the United States.
- Federal, State, and local government policies can influence the rate and direction of technological change.
- Current understanding of this influence and its impact on social and economic factors is incomplete.

ETIP seeks to improve public policy and the policy research process in order to facilitate technological change in the private sector. The program does not pursue technological change per se. Rather, its mission is to examine and experiment with government policies and practices in order to identify and assist in the removal of government-related barriers and to correct inherent market imperfections that impede the innovation process.

ETIP assists other government agencies in the design and conduct of policy experiments. Key agency decisionmakers are intimately involved in these experiments to ensure that the results are incorporated in the policymaking process. ETIP provides its agency partners with both analytical assistance and funding for the experiments while it oversees the evaluation function.

Because all government activities potentially can influence the rate and direction of technological change, ETIP works with a wide variety of agencies, including those that have regulatory, procurement, R+D, and subsidy responsibilities. Programs are currently underway with the General Services Administration, Food and Drug Administration, Veterans Administration, Securities and Exchange Commission, Department of Energy, Environmental Protection Agency, Occupational Safety and Health Administration, and other Federal agencies as well as various State and local agencies.

The accompanying report was prepared from an internal study of system developments conducted within ETIP programs. The purpose of the study was to examine the problems of developing complex systems in the ETIP environment and provide an approach for improved management of those problems. Statements contained in this document represent the views of the author and do not necessarily reflect those of ETIP or the National Bureau of Standards.

Director
Experimental Technology Incentives Program
National Bureau of Standards
U.S. Department of Commerce

NATIONAL BUREAU
OF STANDARDS
LIBRARY

SEP 16 1981

not acc-Ref
QC100
.U56
no. 81-2328
1981

NBSIR 81-2328

A MANUAL FOR DESIGNING AND IMPLEMENTING A PROCESS TO MONITOR COMPLEX SYSTEM DEVELOPMENTS

Stephen D. Garrity

Experimental Technology Incentives Program

September 1981

Prepared by
Experimental Technology Incentives Program
National Bureau of Standards
Washington, DC 20234

Contents

I. Introduction.....1
I.1 How to Use This Manual.....1
I.2 Who Should Use This Manual.....2
II. The Monitoring Problem.....3
II.1 Complex System Developments.....3
II.2 Monitoring a Development.....3
II.3 A Research Role for Monitoring.....3
II.4 The Proposed Approach.....4
III. The Monitoring Factors.....5
III.1 Introduction.....5
III.2 Design Characteristics (Factors 1-12).....12
III.2.1 Operation and Performance Factors.....12
III.2.2 Boundary Factors.....14
III.2.3 Adaptation Factors.....16
III.3 Process Characteristics (Factors 13-24).....20
III.3.1 Introduction.....20
III.3.2 Task Structure Factors.....20
III.3.3 Team Personnel Factors.....22
III.3.4 Project Control Factors.....25
III.3.5 Interaction with the User Environment Factors....30
III.4 User Commitment Characteristics (Factors 25-30).....38
III.4.1 Introduction.....38
III.4.2 System Use Factors.....38
III.4.3 System Support Factors.....42
IV. The Monitoring Functions.....48
IV.1 Function #1 -- Problem Identification.....49
IV.2 Function #2 -- Strategy Development.....49
IV.3 Function #3 -- Research.....50
IV.4 Function #4 -- Documentation.....50
IV.5 Function #5 -- Dissemination.....51
V. A Model for Implementing the Monitoring Process.....53
V.1 Introduction.....53
V.2 Implementation Procedures.....53
VI. Conclusion.....60

I. Introduction

This document is a manual to help system developers design a process to monitor the development of complex systems. It is specifically aimed at projects where developers are faced with substantial uncertainties over design requirements, development processes, and the ultimate home for a system.

The manual is also expected to be of use to researchers of system development processes. By combining research objectives with administrative use of the monitoring process, researchers can conduct structured investigations of system developments in real-time.

The manual is divided into several sections. First, the monitoring problem is identified so that readers can recognize how the proposed process fits into a development. Second, a framework of factors which should be monitored is described. Third, the functions monitoring can serve in a development are discussed. Finally, a general model is provided on how the process can be implemented in a project.

The manual has been derived from an internal study of system developments in the Experimental Technology Incentives Program (ETIP) of the National Bureau of Standards. The study examined ETIP projects and relevant systems literature to determine the important problems in complex systems projects. A monitoring process was then designed for project managers to identify and resolve these problems in a timely manner. Readers interested in examining the details of the study should consult the following document:

> Garrity, S.D. Monitoring system development: A framework and application. Washington, D.C.: Experimental Technology Incentives Program, National Bureau of Standards, September 1981. (NBSIR 81-2327)

I.1 How to Use This Manual

Developers or researchers interested in using the proposed monitoring process may find it useful to scan the manual before beginning

to read the details. It is suggested that all readers review the Introduction section first in order to understand the general problem under discussion and the basic components of the proposed approach. After this, several different readings might be useful.

System developers may want to scan the section on monitoring functions next to determine whether monitoring can enhance their current management practices. Then they should turn to the section on monitoring factors to examine what factors they need for monitoring.

System development researchers may want to scan the monitoring factors first to determine whether use of the factors matches their current research interests. They then should turn to the sections on functions and implementation in order to develop ideas on how to conduct their research within an actual project.

I.2 Who Should Use This Manual

The manual has been developed from a government program where information systems were needed. Thus, to some extent, the monitoring process is oriented toward information systems rather than systems in general.

However, researchers and developers involved in a variety of complex system developments should find the manual useful for their situations.

II. The Monitoring Problem

II.1 Complex System Developments

Projects aimed at developing new systems where there are substantial uncertainties as to system requirements, development processes, and ultimate ownership can present managers with a range of complex, unstructured problems. General models or paradigms to guide system development are of limited use in these cases since circumstances are unpredictable and changing. A process to control development activities as a project evolves is thus a key component for successful management of a complex development.

II.2 Monitoring a Development

A critical need in controlling a complex development is information. Developers must be able to sense when things are going well or poorly and then be able to revise strategy accordingly. For example, if system requirements are initially uncertain, then it would be important to ensure that requirements are established and subsequently reflected in a changing development process. Project managers need to determine whether this is occurring in a satisfactory manner and leading to an acceptable design.

The range of important factors involved and the need for timely action suggests that information be gathered in a structured, routine process over the duration of a project. Managers could use such a process to monitor key areas where problems arise and affect development strategy. The process would provide an early warning of situations requiring management intervention as well as establish a base of information on which to analyze and implement change.

II.3 A Research Role for Monitoring

A monitoring process can also be an important device for researchers of system development processes. Much of the current research in the literature is based on retrospective analyses in which the researcher is either isolated from the project or unable to implement rigorous designs. Systems researchers have a critical need for

more access to actual projects, especially in early stages where the research can be implemented concurrently with the project.

A monitoring process may provide the means by which researchers can gain access to projects. An on-going monitoring system aimed at administrative needs might easily be structured to also satisfy research needs.

II.4 The Proposed Approach

A monitoring process has thus been developed and is proposed as a device which can meet management and researcher needs in a system development process. The process consists of three components:

- A framework of thirty factors which should be monitored.
- A set of five functions which monitoring information can be used for.
- A general model of how the monitoring process should be implemented within a project.

A section is provided on each of these in the manual.

III. The Monitoring Factors

III.1 Introduction

This section describes a framework of factors to be used for monitoring a system development. Thirty factors are identified and they are divided into three categories: design characteristics, process characteristics, and user commitment characteristics.

Design characteristics are factors which describe what a system looks like and how it works. Monitoring these characteristics is important for several reasons. First, the design must be sound, meaning that it actually provides the functions needed by users as well as meet the specific performance requirements. Second, the design must be attractive to users in order to gain their support and use. This means that the design must help users solve important problems in a manner which is matched to their style and other organizational processes.

Process characteristics are factors which describe the methods, structures, procedures, resources, and personnel used to conduct and control the development. Monitoring these characteristics provides developers with an indication of whether process they are using is producing the appropriate design and leading to full implementation of the system at the end of the project.

The third framework category, user commitment, consists of factors which characterize behaviors related to use and support for the system in the prospective owner organization. Commitment is viewed here as developing over time, making it essential to monitor it during the project and determine whether it is increasing or decreasing. The final user decision to fully implement a system will be a reflection of the commitment already established during development.

Two figures are provided to summarize the factors of the framework. First, Figure III.1 lists the framework categories and all of the 30 factors. Second, Figure III.2 provides a brief description of each factor. These descriptions are also used in the more detailed discussions of the factors.

DESIGN CHARACTERISTICS	PROCESS CHARACTERISTICS	USER COMMITMENT CHARACTERISTICS
• OPERATION AND PERFORMANCE FACTORS 1. Response Time 2. Quality 3. Cost of Operation 4. Input/Output Operations 5. Interconnection of Subsystems	• TASK STRUCTURE FACTORS 13. Task Size 14. Task Priorities	• SYSTEM USE FACTORS 25. Applications 26. Consequent Actions 27. Extent of Use
• BOUNDARY FACTORS 6. Capabilities/Limitations/ Expectations 7. User Groups and Their Interrelationships 8. Interfaces with Other Systems and Organizations	• TEAM PERSONNEL FACTORS 15. Skills 16. Turnover 17. Commitment	• SYSTEM SUPPORT FACTORS 28. System Champions 29. Resource Commitments 30. Changes in the User Organization
• ADAPTATION FACTORS 9. Flexible Specifications 10. Matching the System to the User 11. Novelty of the Design 12. Evaluation and Updating	• PROJECT CONTROL FACTORS 18. Organization and Responsibilities 19. Decision Points and Milestones 20. Reports and Reviews	
	• INTERACTION WITH THE USER ENVIRONMENT FACTORS 21. User Involvement 22. Problem Identification 23. Testing 24. Transfer	

FIGURE III.1 THE MONITORING FRAMEWORK

DESIGN CHARACTERISTICS

Operation and Performance Factors

General Description: The functioning of the system design in the user organization.

Factors:

1. Response Time -- The amount of time it takes a system to respond to a user inquiry.
2. Quality -- The accuracy, credibility, and utility of input and output information in the system.
3. Cost of Operation -- The amount of resources needed to operate the system.
4. Input/Output Operations -- The mechanics of user interaction with the system.
5. Interconnection of Subsystems -- The interrelationships of system elements.

Boundary Factors

General Description: The borders of system and subsystem designs.

Factors:

6. Capabilities/Limitations/Expectations -- The conceptual boundaries of a system prescribed by the various groups of people involved in the development.
7. User Groups and Their Interrelationships -- The roles and costs and benefits of involvement with the system pertaining to the different groups of the user organization created by the system design.
8. Interfaces with Other Systems and Organizations -- The relationships between the system and other systems and organizations.

Adaptation Factors

General Description: The match between the system design and the user organization.

Factors:

9. Flexible Specifications -- The match between design specifications and the existing uncertainties over system objectives, processes ownership, products, etc.

FIGURE III.2 BRIEF DESCRIPTIONS OF THE MONITORING FACTORS

10. Matching the System to the User -- The match between the system design and the user organization structure (structural), the abilities, methods, and personal styles of individual users (technical), and the personal relationships of individuals in the user organization (personnel).
11. Novelty of the Design -- The change a new system represents over the existing system.
12. Evaluation and Updating -- The provision of evaluation and updating functions in the system design.

FIGURE III.2 (continued)

PROCESS CHARACTERISTICS

Task Structure Factors

General Description: The design of project tasks to develop the system.

Factors:

13. Task Size -- The amount of resources (money, manpower, and time) assigned and consumed in the performance of individual system development tasks.
14. Task Priorities -- The relative emphasis across tasks at a given point in time and the ordering of tasks over time.

Team Personnel Factors

General Description: The availability and functioning of project personnel.

Factors:

15. Skills -- The availability of technical and interpersonal skills in project teams as required by system development tasks.
16. Turnover -- The change of personnel involved with project teams.
17. Commitment -- The team member support for and implementation of the goals, strategies, and tactics of the system development.

Project Control Factors

General Description: The structures and processes used to control project activities.

Factors:

18. Organization and Responsibilities -- The structure of project teams and responsibilities in the system development.
19. Decision Points and Milestones -- The structure of specific events created by project managers to recognize or review progress and decide future courses of action.
20. Reports and Reviews -- The structure of written and oral communication mechanisms used to document and review development progress.

Interaction with the User Environment Factors

General Description: The involvement and contact between the project and members of the user organization.

FIGURE III.2 (continued)

Factors:

21. User Involvement -- The participation of members of the user organization in the managerial and technical teams of the project.
22. Problem Identification -- The overall amount of attention to and the amount of contact with the user organization in the definition of user problems.
23. Testing -- The amount of concept and design testing conducted in the user environment.
24. Transfer -- The amount of attention devoted to the transfer of the system to the user environment.

FIGURE III.2 (continued)

USER COMMITMENT CHARACTERISTICS

System Use Factors

General Description: User actions related to application of the system in organizational processes.

Factors:

25. Applications -- The use of specific elements of the system in specific units of work of the user organization.
26. Consequent Actions -- The actions resulting from areas of direct system use.
27. Extent of Use -- The amount of use relative to the number of potential applications.

System Support Factors

General Description: User actions related to supporting application of the system to organizational processes.

Factors:

28. System Champions -- The emergence of advocates for the system in the user organization.
29. Resource Commitments -- The type and extent of resources allocated by the user organization to support the system.
30. Changes in the User Organization -- The alteration of policies and procedures in the user organization in order to support system operation.

FIGURE III.2 (continued)

III.2 Design Characteristics (Factors 1 - 12)

The 12 design factors are separated into three major categories: (1) operation and performance factors, (2) boundary factors, and (3) adaptation factors.

III.2.1 Operation and Performance Factors

This category of design characteristics measures how the system operates and performs tasks related to meeting user requests. These characteristics have an impact on people using the system and thus are ultimately linked to the success or failure of the system. Monitoring them during development may decrease the likelihood that inappropriate designs or poor performance stay uncorrected and act as barriers to obtaining full support of the system.

Factor 1 -- Response Time

Description. The amount of time it takes a system to respond to a user inquiry.

Discussion. The time it takes a system to respond to a user, usually in relation to a request, is an obvious design characteristic which will affect use. Long delayed responses might be expected to gradually discourage a user from interacting with the system. Quick turnaround might accelerate introduction of the system and build user confidence. Developers need to monitor the derivation of the response time requirement and its development within the system.

Factor 2 -- Quality

Description. The accuracy, credibility, and utility of input and output information in the system.

Discussion. Quality is used as a measure of input and output information characteristics, especially as to their accuracy, credibility, and utility. This factor is important from two perspectives. First, the system must provide correct information. Second, the system

must produce what is needed by users and lead to their support and commitment to institutionalization. While these attributes can be related, developers must ensure that the system has both since one does not necessarily follow the other.

Factor 3 -- Cost of Operation

Description. The amount of resources needed to operate the system.

Discussion. The cost of running a system is another factor which developers should monitor. If a system is going to cost to much to operate relative to its benefits or user resource constraints, users will be less likely to support it. Obviously, there is a tradeoff between the quality represented in the system design and the economy in operating costs.

It is difficult to know operating costs in advance however, especially if there are great technical uncertainties or little historical data on other systems which might be used as a benchmark. Developers must thus closely monitor costs as a project proceeds and ascertain whether the expected costs of operation are remaining feasible with users.

Factor 4 -- Input/Output Operations

Description. The mechanics of user interaction with the system.

Discussion. The operation of input and output functions of the system is important to monitor for several reasons. First, it must be mechanically easy for users to interact with the system. If it appears too difficult for them to get a request in or out of the system, most likely they will not use it or will find ways to go around it. Second, the same conditions apply to personnel responsible for providing data to a system. It may be difficult to collect the data or there may be uncertainty as to how to put it into the system. These problems could have significant impacts on staff support. Generally, the design of input and output functions may conflict

with or enhance the way an organization works or solves problems. Developers should spend time monitoring this characteristic.

Factor 5 -- Interconnection of Subsystems

Description. The interrelationships of system elements.

Discussion. An important characteristicc to monitor is whether and how elements are connected to each other. Developers should consider organizing subsystems such that interactions between them are minimized. Besides allowing for easier removal and update of components, minimizing interactions may also help minimize the impact of changes to the user organization or personnel relevant to these subsystems.

III.2.2 Boundary Factors

A system can be described by what is included and what is excluded. Similar descriptions can be used at the subsystem level. The boundary of a system is an important concept to developers and its placement can be very influential on development activities.

The main concern in this section is the conceptual boundaries which define the system and the implications they have for the user organization. Conceptual boundaries are the words, thoughts, plans, objectives, expectations, models or roles represented in or guiding design characteristics.

Factor 6 -- Capabilities/Limitations/Expectations

Description. The conceptual boundaries of a system prescribed by the various groups of people involved in the development.

Discussion. Developers need to monitor how well design capabilities, limitations, and expectations match across user groups (and between users and developers). The system's anticipated capabilities should not be oversold; unmet expectations may cause support to whither away or be unattainable. Conversely, user expectations which are

lower than the capabilities claimed by developers may cause the system to appear too grandiose and consequently gain little attention among users.

Factor 7 -- User Groups and Their Interrelationships

Description. The roles and costs and benefits of involvement with the system pertaining to the different groups of the user organization created by the system design.

Discussion. A system usually creates a number of different groups in the user organization. For example, there can be system owners, input sources, output users, system managers, performance evaluators, or system operators. These groups may cut across already established organizational lines.

Another way to identify system boundaries then is to study these groups and their interrelationships. Each group has a specific role in the system and certain costs and benefits associated with it. Some groups may even have multiple roles, such as when owners and users are the same people. Developers need to address both the imbalances a design creates within a group (e.g., between the cost and payoff of participating) and any resulting implications for group interactions (e.g., cost/payoff differences between groups).

Factor 8 -- Interfaces with Other Systems and Organizations

Description. The relationships between the system and other systems and organizations.

Discussion. Various other systems or organizations can interact with a given system. For example, they may be:

- Sources of data,
- Controllers of system resources,
- Users of system outputs, or
- Competitors.

Developers need to monitor the relationships to these outside operations such that the proper ties are made in the design.

III.2.3 Adaptation Factors

Developers need to monitor how well the system and the user organization are fitting together. A poor match should be identified early so that the design and/or organization can be adjusted. In addition, developers also need to integrate the procedures to examine the match into the final design so that they can be used to continually improve the system during its operational phase.

Factor 9 -- Flexible Specifications

Description. The match between design specifications and the existing uncertainties over system objectives, processes, ownership, products, etc.

Discussion. A system design should be kept flexible during a development in order to accommodate (as appropriate) the uncertainties which are likely to arise. Uncertainties may exist from a lack of knowledge about an area (such as a novel technical problem) or might evolve from changes in personnel, organization, or budgets.

In order to keep design specifications flexible, developers should identify and monitor areas that have major implications for the design if they in some way change.

Factor 10 -- Matching the System to the User

Description. The match between the system design and the user organization.

Discussion. As designs are created and detailed, developers need to test the match between the design and the user. This will help determine whether the design and the development strategy are satisfactory.

Three categories of matching seem to capture the major overall concerns:

1. Structural matching.
   This is the match between the structure of the system and the structure of the user organization. Structural characteristics include goals, priorities, patterns of communication, operations, relationships and

responsibilities.

2. Technical matching.
   This is the match between the system design and the abilities, methods, and personal styles of individual users.

3. Personnel matching.
   This is the match between the people in the user organization according to their relationship within the system. Basically, the question is whether people will be able to work together as required in the design.

Factor 11 -- Novelty of the Design

Description. The change a new system represents over the existing system.

Discussion. Developers need to monitor the degree of change a new system represents relative to the existing system. It should be expected that novel changes will be difficult to make or will generate user resistence; incremental change may facilitate conversion to the new system.

Developers need to monitor changes in the following areas and determine implications for development strategy:

- System functions.
  Functions previously unavailable may represent a significant change to users. These functions may have been impossible to develop before, or maybe they were not considered. There may have been no preexisting need.

- Performance.
  Dramatic changes in performance over an older system may represent a source of significant change to users. While the actual performance improvement could represent a small design alteration, the implications of it may be much broader for all associated activities.

- Operation.
  New ways of doing things may be a significant change for users familiar with the older system.

- Other systems.
  The relationship to other systems may change with the implementation of a new system.

- Support.
  The organizational support for a new system might be very different than that given to the old one. A new system may create previously nonexistant barriers to its operation.

Factor 12 -- Evaluation and Updating

Description. The provision of evaluation and updating functions in the system design.

Discussion. In order to maintain the adaptability of a system, developers need to ensure that updating functions are built into a system design. Several different areas need to be examined concerning evaluation and revision processes, including:

- Evaluation criteria.
  Assuming that some kind of evaluation is needed, developers should identify the decisionmakers involved and their criteria. These decisionmakers might include groups well outside of the immediate user units. These criteria may be uncertain in early project stages and may need development. Developers should think through how these criteria can be applied and built into a design.

- Decision points.
  If explicit decision points are used to control a project at each stage (see also Decision Points and Milestones below), developers should examine early what information might be needed about the system. Implications from analyses like these may possibly provide the basic strategy for later system testing (see also Testing, Factor 23).

- Other organizational review processes.
  If the system is to become an integral part of an organization, its evaluation and updating functions may need to be tied to already existing organizational review procedures. These might include reviews of the organization conducted from the outside as well. Developers may have to adjust designs to accommodate changes from these sources.

- Freezing a design.
  At some point it may be advantageous to freeze a design in order to bring the full system into operation without continuing distractions for revisions. Developers need to examine where updates should be made and which ones can be delayed without risking problems. In cases where updates are put off, provisions should be made for users to initiate them when they assume complete responsibility for maintenance of the system.

- Mechanisms.
  Finally, developers need to examine how evaluation and updating function overall. They should study whether problems are identified, whether redesigning occurs in a timely manner, and whether changes actually are, or can be, made. This mechanism must be operational like any other system component when the system is finally institutionalized.

III.3 Process Characteristics (Factors 13 - 24)

III.3.1 Introduction

A system development can be described by the methods, procedures, personnel, resources, and structure developers use to operate the project. These factors contain a process perspective in that they are a means towards an end: the eventual design and institutionalization of the system. Factors which denote project activities like these are defined as process characteristics in the monitoring framework.

The following sections identify and describe 12 factors. These have been separated into four categories: task structure, team personnel, project control, and interaction with the user environment.

A special note should be made here concerning the different groups which may be involved in the development. A development may be the responsibility of several different institutions which are normally independent of each other. For example, contract support may be needed or a third party may provide financial support.

The development process under multiple group arrangements like these can be complicated. Responsibilities are shared, the different groups must be coordinated, and different perspectives on strategy may arise.

For this reason, some of the following sections for each factor include discussion on how multiple group situations can affect a project, particularly in contracting cases. These discussions illustrate the additional characteristics developers should monitor when more than one group is involved.

III.3.2 Task Structure Factors

A system development can be divided into a series of specific tasks or group of tasks. Tasks are usually grouped into categories or phases, such as problem identification, design, or evaluation. These demarcations are fairly arbitrary, however, and disguise the true cyclic process and sequencing of development tasks.

Of more interest are the actual tasks which occur within and

across phases like these. Identification and coordination of these tasks represent a significant problem to developers. For example, tasks related to different subsystems may be similar and need coordination. In other parts of the design, some subsystems may need to be fully developed before others.

Developers need to monitor both the size and priority of tasks as the development proceeds.

Factor 13 -- Task Size

Description. The amount of resources (money, manpower, and time) assigned and consumed in the performance of individual system development tasks.

Discussion. The size of project tasks, or group of tasks, is one characteristic developers should monitor throughout a project. The size of a task includes the amount of money, manpower, and time consumed in performance of the task.

One of the main reasons to monitor size characteristics is to control costs. Costs can easily expand beyond initial expectations if there are major technical uncertainties in the design or numerous changes of course in the project. Early and progressive monitoring of resource expenditures may help avoid becoming overcommitted to a narrow portion of the project or running out of resources at key points.

While there is no one right size for project tasks, many sources recommend dividing projects up into small, manageable steps which are easier to control. There are several reasons behind this strategy:

- Overall, system developments are complex undertakings which involve numerous, intertwined problems. Breaking off and working on small pieces of the problem can make the project easier to handle. As more is learned about the user environment, more complex and expansive tasks can be undertaken.

- In early stages of a development, smaller investments of resources by users and developers can reduce the risks involved in choosing one direction over another. The chances of a large scale failure may then be reduced.

- Developments are likely to contain a number of changes which can upset the flow of the project. Using smaller, discrete

tasks and gradually moving towards more complex undertakings can help protect against major disruptions.

- Overall success may be more easily attained when smaller lines of work are used. A strong track record of modest success may increase the likelihood of overall success.

On the other hand, smaller, more numerous tasks imply more project control and monitoring than might otherwise occur. This could become most inefficient if carried to an extreme. Developers have to balance the risks in commiting themselves to certain task sizes with the benefits they can achieve in work flow, control, and likelihood of eventual success.

Factor 14 -- Task Priorities

Description. The relative emphasis across tasks at a given point in time and the ordering of tasks over time.

Discussion. The priority of tasks is another characteristic developers should monitor throughout a project. Priorities include both the emphasis across tasks at a point in time and the ordering of tasks over time.

There are a number of factors which can affect priorities and these should be considered when priorities are monitored. These include:

- Resource availability (When will money and manpower be available for a task?).
- Time needed to accomplish a task (Is a task uncertain or well understood?).
- Priority of user needs (What should be accomplished first from the perspective of the user?).
- Interrelationships of subsystems (In what order must subsystems be developed?).
- Need for accomplishments (What tasks will help build momentum for the project and build credibility with the user?).

III.3.3 Team Personnel Factors

A second process characteristic developers need to monitor is

the operation of the system development team.

Developers must first ensure that the skills needed in the project team are obtained. These will range from managerial to technical areas.

Second, developers must monitor the turnover in personnel that is likely to occur. Turnover of key personnel can slow progress in a development by consuming time to transfer knowledge between incoming and outgoing personnel, or by bringing in new perspectives which must be accommodated with already existing strategies.

A final area developers should monitor is the commitment of personnel to the project. Commitment of the staff to a project is a predecessor to the more comprehensive commitments by users. Changes in these commitments, or differences between team members, may be an early indication of uncertainties in project goals or stategy.

Factor 15 -- Skills

Description. The availability of technical and interpersonnel skills in project teams as required by system development tasks.

Discussion. System developments require a range of skills which developers must ensure are available. At one level, there are the technical aspects of design which require people trained in specialized areas. At a second level, there are interpersonnel aspects of design which require people trained in understanding and working with people. A development team should have a mix of these skills. It is also important to have people with a mix of both technical and interpersonnel skills who can effectively handle the complex, intertwined problems that developments involve.

Contracting for these skills may be necessary, and this may present an additional number of problems developers should consider. For example, contracting requires developers to design a procurement procedure that specifies the kinds of skills needed. In addition, contracting for support results in a new, outside team that requires

integration into the user organization.

Factor 16 -- Turnover

Description. The change of personnel involved with project teams.

Discussion. A major problem for a development team is turnover in its personnel. Important knowledge can be lost in key project areas that ensures continuity. Even in cases where no knowledge is lost, newcomers will need time to familiarize themselves with a project. This can cause delays.

Developers might prevent some turnover by initially obtaining long term commitments from key team members who would represent a significant loss to the project. If contractors are involved with the project, this is often done through the use of key personnel clauses in the contract.

Changes are inevitable, however, and developers need also to examine ways to minimize their impacts during a project. Several actions may be taken:

- Developers can establish an executive management team, and other teams for that matter, such that key tasks and responsibilities are shared among several people.
- Another method is for developers to continuously encourage people to remain throughout a development.
- Developers may want to minimze broad assignments for project personnel.

Factor 17 -- Commitment

Description. The team member support for and implementation of the goals, strategies, and tactics of the system development.

Discussion. Commitment of project team members to the goals, strategies, and tactics of the development is an important predecessor to the eventual full user commitment to the system. A lack of personal commitment may result in key staff turnovers (see preceding section),

or be evidence of a much broader problem in the home organization. Gaps in commitment between team members may be a source of confusion and uncertainty which slows progress. Gaps between team members and their home organizations may have the same effect. Developers (team members) need to be sensitive to the commitments of personnel in the project. Shifts or gaps need to be detected early and their broader implications handled in line with project goals.

Several indicators of project team commitment are the following:

- The involvement of key, experienced personnel on project teams from the various groups participating (e.g., from the user organization, contractors, etc.).
- Top level management support from the home organizations.
- Time spent on the project versus time spent on competing projects in the home organization.
- Differences in commitment between personnel of separate home organizations.

III.3.4 Project Control Factors

Several factors having a significant effect on a system development are the processes used by developers to control the project. Large scale system developments are complex undertakings which involve numerous activities, problems, technical issues, opportunities, etc. They can be further complicated in situations where more than one organization is involved in the project. Developers need to monitor the control processes used so that revisions can be made if they prove ineffective for accomplishing project objectives.

Three categories of process control methods are discussed in this section. The first is the organization of teams and responsibilities. Developers need to establish a management structure which matches the complexity of the development and the appropriate level of distributed responsibility. The second category is the use of decision points and milestones. Developers can use specific points in time or the accomplishment of selected tasks as control points where progress can be reviewed, problems can be identified, or approv-

als for further work can be obtained. The final category is the use of reports and reviews to monitor work, document and discuss problems, or to disseminate information about the project to relevant outside stakeholders who may have influence over the development.

Factor 18 -- Organization and Responsibilities

Description. The structure of project teams and responsibilities in the system development.

Discussion. A key part of project control is the team structure used by developers to manage and conduct the project. Part of the structure should include a central authority which can coordinate and manage the numerous organizations and people involved. Another part of the structure will involve the distribution of responsibilities to the personnel actually performing the work. Day-to-day control of project activities will be located at lower management levels. Developers will need to examine early in a project the kind of structure they use to conduct and control the work and then progressively monitor its effectiveness in accomplishing project tasks.

Many sources advocate the use of a small central team for managing a systems project. This team controls development overall and reports to higher level decisionmakers in the organizations supporting the project. In projects where only one organization is involved, such as with an in-house development, the team should probably consist of both users and designers with users retaining ultimate control.

However, the situation may be more complex when mulitple organizations are involved, such as when contractors are used. While central authority should still reside with the user organization, much of the project work may occur outside of the user organization. Thus, to some extent, control must be more distributed. A structure must be established that provides a locus of control while ensuring adequate responsibility to outside groups.

Another issue is whether developers utilize new teams or existing ones to form the working level structure. Developers may choose

to form new teams to conduct the work by drawing on personnel from existing organizational units. An alternative is to rely on already existing groups or structures. In this case, project tasks would be assigned to the units who would then decide how to perform the task and staff up for the work.

Whichever approach is taken, developers will have to examine how responsibilities are delegated to the working level teams. Centralized control of all project activities will likely be impossible, and for that matter, inefficient and unnecessary. Numerous problems and decisions can be delegated to lower level working groups where they can be effective handled as they arise. It will be advantageous overall if working level groups can gain the sability to recognize problems and opportunities on their own and then have the flexibiity to handle them as appropriate for project objectives.

It should be recognized, however, that some problems (or opportunities) may arise which fall outside the organizational lines chosen by developers. Developers should consider the need for ombudsmen who can be assigned to these unique problems. These individuals might report directly to the central management team.

Factor 19 -- Decision Points and Milestones

Description. The structure of specific events created by project managers to recognize or review progress and decide future courses of action.

Discussion. Large scale developments involve numerous activities, problems and opportunities which need some level of ongoing recognition and approval from the central management team. Developers first need to examine in advance where major and minor decision points should be placed and then actively monitor their occurrence.

Major decision points should be located at the end of each phase to mark the conclusion of the activity and to select the strategy for the next phase. They can also protect against making commitments too early (or the fly before buy strategy).

The exact placement of major decision points is fairly arbitrary, however. Activities implied by the name given to a particular phase probably recur throughout the development of a system, particularly if the progress on different subsystems varies or if cycles are made to refine earlier designs in light of new information about the user. Developers may have to select points at which the development is, in general, making a transition.

Major decision points at the end of phases are not the only control points needed in a project. Developers will also need to emphasize sequential approval and acceptance of progress during each phase. Problems and opportunities can be identified more quickly and acted on at a point when action is needed. Reworking, redesign, and argument after the fact can be avoided.

Within each phase, developers should thus select events or milestones as formal, recognized points for monitoring. Milestones can be either points in time or events. They might represent a series of events which conclude with some identifiable accomplishment. The key is for developers to select milestones which provide insight to project activities and prediction power for future sucess or failure. Milestones should be:

- Objective, in that they are not subject to widely different interpretations,
- Material, meaning they can be seen directly, and
- Significant, in that their accomplishment has some significance in the project.

A series of milestones can have the additional benefit of leaving a better documentation trail and improving the decisionmaking conducted periodically at the end of phases.

Factor 20 -- Reports and Reviews

Description. The structure of written and oral communication mechanisms used to document and review development progress.

Discussion. Decisionmaking and control of a complex system development will involve some level of information transfer between the various groups involved. Transfer and liason with other groups outside of the project may also be needed. Two common mechanisms for processing and transferring information are reports on project activities and periodic reviews of major technical and organizational issues. Developers need to structure in advance where these reports and reviews occur and who will be involved. Developers then need to monitor whether these mechanisms prove to be sufficient for transferring key information of the level and at the timing needed to control the project.

Developers need to consider several factors when designing or scheduling written reports. First, reports require a significant amount of time to prepare. They can distract key members of the development team from the priorities of the project. Developers should keep the number and extent of reports small and monitor whether too much effort is being diverted to their production.

Second, reports may serve several purposes beyond project control which need to be considered when reports are designed. For example, developers may want to document portions of the project for later review. This might be important in areas where uncertainties remain and developers choose to forego further work; later exploration could be more fruitful. Reports might also act as marketing documents for the project and the system.

Third, developers need to consider the appropriate level of reporting. In some cases, more informal liason between groups may serve the purpose of transferring the needed information, while in other cases more formal communication through channels is needed.

Finally, since reports take time to prepare and revise, their utility in real time project control may be limited. Developers may need to rely on other means to help control the project (see design reviews below) and let reports act more as a record of already recognized and approved activity.

More direct real time control may be exerted through the use of face-to-face meetings and discussions which can quickly pinpoint problems and options. It is common practice in system developments to use design reviews for the purpose of reviewing general problems, strategies, and the like, and making changes.

The basic strategy of these reviews is to compare progress, either in specifying a design or actual construction, with previous design plans. Inspections, for example, mean comparing hardware with the paper design. Demonstrations can mean the review and acceptance of assembled subsystem components in comparison to expected overall system performance specifications. Reviews like these could also be extended to more narrow, specific problems or subsystems and occur more frequently than once during a phase. Developers need to identify what reviews are useful and who should participate and contribute to them.

III.3.5 Interaction with the User Environment Factors

A key part of system development is the interaction developers have with the user environment. Interaction is needed in order to transfer information about the user problems to the design teams and to progressively develop and transfer the system to users. Interaction is also beneficial for establishing relationships with users. Their openness and support will be critical for design and eventual institutionalization.

There are two related perspectives developers should monitor about interaction. First, there is direct user involvement in the project. User involvement is usually one of the main factors found to be important to project success. Second, there is contact with the user organization in general. Extensive contact should be made in order to extract the key user problems around which the system needs to be designed, verify concepts and system performance, and to transfer the finished design. Sections on each of these processes are presented below.

Factor 21 -- User Involvement

Description. The participation of members of the user organization in the managerial and technical teams of the project.

Discussion. One of the major factors found to be important in system development is the involvement of users. Projects conducted in isolation from users have often resulted in designs which do not meet user needs and gain little support for institutionalization. Developers need to identify areas where user participation is required or beneficial and monitor tasks to ensure that participation is solicited and utilized.

One approach to user involvement is to include users on project teams, both at the managerial and working levels. This provides a mechanism for user control of the project and a basis for eventual ownership of the system. It also helps ensure access to the organization for the collection of information needed by design teams.

Developers should also consider involving users from different levels of the organization. Managers and line groups may have different perspectives on the information problems and the appropriate designs.

Obtaining user involvement may be a difficult problem in itself, however. Users may not have the time to be actively involved or it may be unclear who should be involved, particularly if the system is a new entity for the organization and the final home is unknown. If there is little concensus on the need for a system, users may also resist involvement and continue to support other systems.

Close involvement may also have some risks which developers should consider in soliciting participation. Early project stages are likely to be characterized by exploration, uncertainty, and shifting concepts. Utilizing the limited time of users at this stage may be inefficient until clearer tasks emerge. In addition, high levels of involvement may raise expectations about the system. Difficulties or delays may lead to higher disappointments later if these initial expectations are not met.

Factor 22 -- Problem Identification

Description. The overall amount of attention to and the amount of contact with the user organization in the definition of user problems.

Discussion. One of the important parts of the development process is problem identification. Information problems are likely to be unstructured and complex. They will probably be closely intertwined with other problems in the user organization. Early definitions of the problem may be too general or simplistic and, in any case, will probably change as more is learned about the user environment and design options. Attention to problem definition should be given early and routinely throughout a project.

A systematic approach to problem identification will benefit the development. Emphasis should be placed on learning about the user environment from a number of different perspectives: how it works, goals, decisionmaking styles, decision processes, information flows, other information systems, etc. The goals and expectations for a new system would also be an important item to include here. Developers should conduct analysis at the highest decisionmaking levels and work into the organization, while concurrently examining the lower working levels and moving up decisionmaking chains.

Problem identification should also be extended to the external environment of the user. The user organization will probably have relationships with outside groups that can have significant effects on the organization itself. Developers should examine and monitor these linkages and determine how they might affect system design (see also Boundary Factors above). It would also be beneficial to determine what information about the system these groups might need during the development (see also Reports and Reviews, Factor 20).

As information is gathered from these areas, it may become useful to organize it by constructing some models. At first these might be descriptive models about what is happening: the players, the decisions, and the information needs. Later, a second set of normative models can be developed to describe the decision processes users would like to have. Descriptive and normative models can then be compared

to help identify key information problem areas.

Developers should also consider how the information they collect during problem identification can be used to help the organization directly. A system development may present one of the first times the organization has been closely examined, particularly by outsiders. Developers may uncover factors about the organization that were not previously known. Detailed feedback to users about current processes may identify improvements which can be made immediately without proceeding with a new design. Close study may also reveal widely varying knowledge or perspectives on organizational problems, including the information problem of concern. It may be important to project success to reduce these differences. User recognition of the need for change has been found to be a key factor in successful developments. Developers should spend time communicating and selling the problem to the user.

Spending time on problem definition with users may also have additional benefits besides locating the need for a system. It will begin to set a precedence for close user contact and involvement with the development. Some users may not understand what a systems project is or will have significant anxieties over how it will change their jobs. They may resist providing information or access to their programs. Similarly, developers may be uncertain of what will happen or be inexperienced in what to do. Close contact between these two groups can help reduce these problems by promoting a mutual understanding.

Other benefits might include the following:

- Establishing a language about problems and designs which users and developers understand.
- Identifying key underlying assumptions in problem definitions which need examination.
- Discovering areas where resistance to change may be great and organizational politics important.
- Starting to build user commitment to the system.

Factor 23 -- Testing

Description. The amount of concept and design testing conducted in the user environment.

Discussion. As more is learned about the user information problems and the user environment, concepts and designs for the system will emerge. Eventually, it may become necessary to directly test designs in the user environment as a means of substantiating design claims. Testing in the user environment may also be important and beneficial for:

- Competitive demonstration.
  Several competing proposals for a system design may emerge in the project and testing in the user environment may be critical for selecting among them.

- User involvement.
  Testing in the user environment can give users something to react to, especially those users not directly involved in project teams. Their knowledge and opinions may be critical factors for system designs. Participation in a low risk test may be an effective means of obtaining their ideas and support.

- Commitment.
  Favorable results from tests may be one important factor which eventually results in full user commitment to the system. In addition, opportunities may arise for early use of the system while it is still under development. These may provide an early payoff which should be taken advantage of.

Developers should establish an active program of testing with users and continue it as long as necessary in order to resolve uncertainties in design.

There are several levels of testing in the user environment which might be considered by developers. One is to use models on paper or dummy mock-ups of the system and/or its products. The earliest of these might be descriptive and normative models of user decision processes which evolve in problem identification activities (see above section). Rapid feedback to users of information like this can further educate users as to what is needed as well as provide a test to determine whether developers are understanding their needs.

At another level, working models of subsystems might be constructed and pilot tested under various conditions with users. Different

parts of the overall design can be tested by themselves and at the appropriate stage of development. For example, developers might choose to focus on a particular decision problem and then produce a working model to test their design concepts for this one problem.

Finally, as many of the system components become developed, a prototype system can be formulated and tested with users. A prototype design would be expected to include most or all of the functions developers had found to be needed by users. It would also include the various interconnections between subsystems or between the system and outside groups. The prototype design would provide for a full scale system test and check the emerging design against the original performance objectives.

An important part of testing at any of these levels will be the development of user criteria to evaluate the performance of designs. Developers will need to identify in advance and progressively refine the criteria users prefer to apply. Some of them may be appropriate for institutionalization within the system for routine use after development is completed (see also Evaluation and Updating, Factor 12).

Factor 24 -- Transfer

Description. The amount of attention devoted to the transfer of the system to the user environment.

Discussion. A key transition point in a systems project is the transfer of control and support from the developer to the user. Transfer of the system to the user is often referred to by several names: implementation, institutionalization, cut-over, or conversion. Developers need to think through the implications of changing to a new system and routinely monitor whether their strategy is leading to a smooth transition of ownership and full user commitment.

Transfer of the system to the user is an activity which can and should begin early in a project. As concepts and pieces of the design evolve during problem identification activities, developers

should consider how the designs would become fully operational and institutionalized within the organization. Examining the differences between the new design and existing practice will help identify possible transition strategies and key problems (see also Novelty of the Design, Factor 11). Close user involvement in this analysis is also important, both to identify issues and to initiate the transfer of concepts.

Later as designs are detailed, testing provides another stage in which transfer considerations should be studied by developers. Tests in the user environment may be designed to include user participation in using and working the system. This will provide a temporary means of exploring how to transfer system components to users from the developer perspective. It will also provide a low risk means for users to explore the transition to a new system. Feedback from their perspective will be useful. Developers should be able to learn lessons about transfer from testing which can be used later in future tests and final institutionalization.

When it is appropriate or required, the system or individual components will be transferred permanently to users (the turnkey system). Developers may have the choice of transferring the system gradually or quickly making the transition from the old to the new system. Several factors can be considered in deciding which approach to take:

- Abrupt transition.
  Complete rapid transition to a new system may be too risky or unnecessary. If there are user problems needing quick attention, it may be beneficial to institutionalize key components as they are available. Radical changeover may create a number of side impacts with users that only complicate the transition and start-up periods.

- Backup systems.
  It may be beneficial to gradually phase in the new system while gradually phasing out the old. This will provide some backup or protection should unexpected problems occur. Users should not be left without any system if delays occur.

- Training.
  Users need to be trained to work with a new system. Users should be involved in developing training programs and training needs should be a part of design activities. The availabil-

ity of these programs may influence the transfer point.

- Documentation.
  Documentation of the design and related matters should be complete, especially before transfers are made. Areas where design problems remain should be thoroughly documented so that users can continue development later.

- Management support.
  As the point for transferring a system to users approaches, it is possible that managers (both of the project and the user organization) may start assuming that the system is complete and its implementation routine. This may cause a lack of attention towards transfer which jeopardizes final stages. For example, lower level and less experienced staff may be assigned to transfer tasks and cause implementation failure.

When the system has been mostly or completely transferred to the user, a final stage of withdrawal and termination should begin for the project teams. In termination, developers need to ensure that:

- Ownership and control of the system rests with those who must use and maintain it.

- Necessary new patterns of behavior have become a stable part of the user's routine.

The basic goal is to refreeze the organization by removing the disturbances of change and leaving the organization in a stable position. Developers should ensure that evaluation and updating functions become operational so that users learn to revise the design themselves (see also Evaluation and Updating, Factor 12).

III.4 User Commitment Characteristics (Factors 25 - 30)

III.4.1 Introduction

The final category of factors in the monitoring framework is for measuring the extent of user commitment to the system. Commitment to the system is viewed as a series of user decisions or actions over time which indicate increasing user interest, support, and acceptance of the system. Full commitment to the system will emerge when users elect to assume total responsibility for the operational system and the development project is terminated.

This section identifies and describes six factors for developers to use in monitoring commitment. These have been divided into two areas: factors relating to use of the system and factors relating to support of the system.

III.4.2 System Use Factors

Major indicators of user commitment to a system are the actions taken toward use of the system. In early project stages, these user actions may be decisions about the design and application of the system to organizational processes. Later, as the system is developed, these actions will change to use of the prototype system or subsystems. Developers need to closely monitor these actions so that changes can be made in the design or the development process if needed.

Three factors are described in this section. First, developers need to identify the types of applications of the system (or ones it is expected to have). This requires identifying system actions with specific user tasks. The second indicator of use is the activity resulting (or expected to result) from system applications. These actions need to be documented as they will likely play a key role in obtaining user commitment to the system. Finally, developers should monitor the extent of use in order to gauge how frequent and widespead it is (or will be). This may be an especially important factor to the top level managers who will decide whether to provide the resources needed to fully support and institutionalize the system.

Factor 25 -- Applications

Description. The use of specific elements of the system in specific units of work of the user organization.

Discussion. Developers need to identify the applications of a system to user problems as one part of monitoring user commitment to the system. In early project stages when designs are being formulated, developers will have to rely on the expectations and plans about applications rather than actual cases (see also Boundary Factors above). These plans can provide important insight, however, to the extent and type of commitment users expect to make. Later, as system components become operational in testing and implementation stages, developers can then monitor whether the applications are in fact occurring and are leading to full institutionalization of the system. Problems or changes in applications may indicate a need to revise the design or project strategy.

It is important that developers take a broad view of what constitutes an application. This primarily means that definitions of use should not be restricted to simply identifying whether a major decision is the result of system use. Numerous applications of the system may have occurred to support a decision process even though a major decision is not evident.

Developers should also include unexpected applications and areas of non-use. Users may perceive that the system will not (or actually doesn't) solve their problems. They may turn to other systems and withdraw their support from the project. Developers may also find that staff who operate the system are having difficulty doing so. They might prevent or seriously impair further operation in an area unless the design is changed or they receive more training. User resistance at these levels may have serious implications for eventual user commitment to the system.

The key problem in determining where applications occur (or will occur) is to match some unit of work in the organization, such as a decision, with some unit of system action. This matching can be greatly facilitated by the use of models which detail the informa-

tion and decision flows of the user organization. Developers can take these models and identify places where the system interacts with the user. The range of applications can then be determined as can the specific user behaviors which define work tasks.

It is advantageous to start modeling early, such as during initial problem definition stages when extensive contact with users occurs (see also Problem Identification, Factor 22). Developers can then:

- Obtain an early indication of commitment by matching expected system products and applications. This can be compared to project goals (see also Boundary Factors).
- Plan the occurrence of applications as the project proceeds, possibly starting with simpler ones and then moving into more complex undertakings (see also Task Structure Factors).
- Obtain a basis for establishing measures of actual use which can be applied later during testing and implementation (see also Project Control Factors).
- Uncover any different perspectives as to what constitutes use. Differences among users may prevent agreement that an application has occurred (or will occur) and have an effect on eventual institutionalization (see also Boundary Factors).

Factor 26 -- Consequent Actions

Description. The actions resulting from areas of direct system use.

Discussion. Further indicators of use are the user actions which come about as a consequence of system applications. These actions should be observable user behaviors which can be documented and shown to others. Changes in thinking may be a valuable result of system use, but these are hard to measure and probably would be considered weak evidence to justify support for the system. Credible evidence is of critical importance to both users and developers. User managers need proof that a system improves the operation of their organization before they provide full support. Developers need to evaluate performance and commitment and then determine whether changes are needed in design or strategy.

There are several categories of actions developers might consider monitoring in order to measure the effects of system use. These include:

- Follow-on actions in organizational processes.
  Developers can identify the linkages of various organizational actions and trace the impact of system use down the line.

- Organizational changes.
  Use of the system may eventually bring about changes in the organization, such as in structure or official procedures. Changes may occur in other systems or processes which support system use (see Changes in the User Organization below).

- Performance changes.
  Factors which are used by the organization to measure performance might record changes which coincide with the introduction of the system. Performance improvements may consequently lead to high level commitment to the system.

- Avoidance or non-use of the system.
  Negative experiences in a particular application may lead some users to avoid further contact with the system, discontinue their support, or work against the system (see also System Champions below).

As with identifying system applications, determining where actions like these occur and then measuring them can benefit from initial detailed modeling of user processes. Models can help identify the flow of information or decisions and provide developers with a means of tracing user actions which emerge from specific system applications. Identifying these actions in advance also provides a means of determining what decisionmakers consider valid evidence of use. In addition, expected actions identified early through modeling can be compared with actual behavior during use and provide a means of measuring changes in user commitment.

Factor 27 -- Extent of Use

Description. The amount of use relative to the number of potential applications.

Discussion. The third indicator for monitoring use is the extent of system applications to user processes. Measuring extent

includes determining the frequency of use and the amount of use relative to the number of potential applications. These factors should be monitored over the life of the project and used to indicate changes in user commitment. Stable or decreasing levels of use might indicate dissatisfaction with the system design or loss of momentum in project strategy. Revisions in both areas may be needed.

One major question that developers and users alike have to consider is how much use constitutes acceptance of the system. User top management will likely have some threshold for the extent of use above which they will give serious consideration to full support and institutionalization. This threshold may be based on how often the system is used over the range of different applications as well as how this use affects the performance of the organization. Developers can begin to uncover thresholds like this early by spending time detailing the types and extent of applications decisionmakers expect to achieve. This may be best done by, again, modeling user processes where the system is to operate and determining what use specifically means. Developers should then continue to examine the extent of use as designs are implemented and determine whether the expected levels are achieved and prove to be significant enough to justify full institutionalization.

III.4.3 System Support Factors

Besides monitoring the direct use of the system to determine user commitment, developers should also monitor other user actions which indicate support for the system. These actions may include early signs of user acceptance such as the emergence of user advocates as well as longer range institutional changes which are needed to support the operation of a system. The lack of supporting actions like these may indicate that the system is isolated from the user organization and that it will not become fully implemented when the development ends.

Three factors for monitoring system support are described in this section. These include the emergence of system champions, the

resources allocated by the user during development, and the changes in the user organization which are needed to support system use.

Factor 28 -- System Champions

Description. The emergence of advocates for the system in the user organization.

Description. An important source of support for a system development can be a critically placed key man, advocate, or system champion. A system champion is a person, perhaps best located in the user organization, who believes very strongly in the system: its concepts, design, performance, etc. The system most likely matches the champion's objectives and the organization's objectives. A champion has credibility inside the organization, hopefully across different levels. His credibility may even extend outside the organization as well. He can push the system into existence and also find the pull for it from users. He should also be able to isolate the system from attack by others.

Developers should seek out system champions and cultivate their participation. Their help can be a critical element for success throughout a project, especially in early stages when both the project and the system are being defined. System champions may offer the earliest sign of user commitment and act as a catalyst for acquiring user support.

Champions may emerge from different levels of the user organization and developers should actively seek this broad-based support. Champions from top levels of management are clearly important to a development since they may have great influence over resources, access to the organization, or project strategy. However, developers should seek champions at other levels as well. This can help gain credibility for the system with key groups and also expose project teams to different perspectives or expectations about system design or project strategy. In addition, multiple champions, either within or across organizational levels, may reduce the potential disruptions caused by the loss of any one of them.

Developers must be skeptical, however, of champions who may be overly supportive or controlling. They may be attempting to capture the project, and thus the system, in order to use it for their own personal rather than organizational objectives.

Developers must also be aware of those who actively oppose the system. Just as the new system may have a key advocate, it may also have someone who is uninterested or critical. Opponents may fight development at each stage or build up opposition. They may also elect to hamper an operational system as much as possible or sabotage system processes. Developers should be open to the possibility that opponents exist, or will develop, and be prepared to actively counter the effects of their actions.

System opponents may not be limited to those who oppose development outright from the inital planning stages. Technical problems, infighting already existing in the organization, differences of opinion over designs, or opposition to particular project decisions may create new opponents during a development. In these situations, it is critical that developers direct specific attention to the resolution of conflicts and consensus building. System developments can involve change and disruption to users, and the creation of an active opposition only makes strategy that much more complex and success that much more difficult to achieve.

Factor 29 -- Resource Commitments

Description. The type and extent or resources allocated by the user organization to support the system.

Description. A key indicator of user support for a system is the type and extent of user resoucres committed during the development. The two primary resources users can provide are manpower and money. In early development stages, these might be supplied to the project to support design and testing. Later, as subsystems become available for implementation, user resources may be supplied to operate them. Each of these allocations is a sign of user commitment to implement a system after the development is completed.

There are several characteristics concerning allocation of money to a project which developers may want to monitor. These are:

- Temporary versus permanent allocations.
  In some cases, part or all of the development may be funded out of special user project funds and be combined with resources originating from outside third parties. A change in budgeting, where the system becomes a line item in a user budget, may be an important indicator of emerging support.

- Origin of money.
  Developers may want to compare the origins of financial resources with the location of primary users. Arrangements not following organizational lines could mean broad support or potential ownership problems. In addition, if multiple sources are involved, developers may want to consider the relative proportions of allocations versus expected use. Continuing support from all parties may be essential for full scale system implementation.

- Planned allocations/expenditures.
  Developers may want to examine how user allocations are planned over time and in what system areas they apply. Targeted resources may signal important areas where user commitment will be based. User expenditures should also be monitored. Success in the development will most likely be followed by increasing resource allocations.

These factors may not apply in projects which are entirely based on third party funding and where user resources are expected only after successful development.

Similar characteristics can also be monitored for user allocations of manpower. They may apply independently of any funding allocations and be essential for strategies involving close user involvment. Developers should consider:

- Temporary versus permanent assignments.
  User staff may only be temporarily assigned to project teams or to the operation of initially available subsystems. While these conditions would indicate an initial commitment, a transition to permanent assignments to operate the system components would indicate a more significant level of support.

- Type of personnel assigned.
  The development may require the skills and involvment of users, and developers should monitor whether they are made available. The assignment of key people in an organization may signal

a high degree of user commitment to the system (see also Team Personnel Commitment, Factor 17).

- Level of involvment.
  A key sign is also the extent of involvement. User personnel assigned to work on the system a small part of the time may be too distracted by other business to contribute effectively to project objectives. Allocation of significant portions of staff time may indicate a solid commitment to the system. Commitment would also be indicated by increases in the proportion of their time.

Factor 30 -- Changes in the User Organization

Description. The alteration of policies and procedures in the user organization in order to support system operation.

Discussion. Other indicators of user support and commitment to a system are the changes which take place around the system. Systems are placed into complex organizations involving intertwined line of communication, support, responsibility, and activities. Changing over to a new system likely causes and/or requires changes in other places and sytems. New relationships and functions may be established. Developers should look beyond the immediate areas of system impact to other areas where change may indicate a positive or negative force for user commitment.

There are numerous potential changes inside the organization which developers can monitor. Some may occur in the areas which are to support the operation of the system. For example, the system may require new types of personnel or new positions which must be arranged for by personnel divisions. Some users, may need training in order to operate the system and new programs may be created for this. Another similar change is the creation of an incentive system either to attract users to the new system or to acknowledge improved performance because of its use. Other changes may occur in processes, such as decisionmaking style or how an organization works. Changes may also be observed at higher management levels. For example, the system may support a new policy which high level decisionmakers must acknowledge and support. Finally, developers should examine changes

in competing systems. The continuation of competing systems may indicate a lack of support for a new system, while conversely, the gradual withdrawal of support from them may indicate the opposite.

Developers should also monitor actions or conditions outside of the user organization which may affect user commitment to the system. For example, outside institutions may mandate the use of the system. Outside organizations may also have a role as system users or suppliers of information. Changes in their roles or activities may be a direct or indirect sign of system acceptance and support in the user organization.

IV. The Monitoring Functions

The following sections describe the five functions monitoring can serve in a system development. As described earlier, the monitoring process has been designed to serve both administrative and research purposes.

The five functions have been designed to overlap and support each other for these uses. These functions are:

- Problem identification -- The tracking and assessment of key areas where development problems typically occur, both in the short and long range.
- Strategy development -- The identification and development of explicit management actions to solve problems and revise strategy.
- Research -- The design and implementation of real-time studies of the development process.
- Documentation -- The establishment of an organized and stable recording process which can identify special or recurring problems, and support decisionmaking and research.
- Dissemination -- The distribution of key development information to these inside and outside a development as a means of facilitating coordinated actions and distributing knowledge gained.

IV.1 Function #1 -- Problem Identification

Problem identification is the major function for the framework, reflecting the original government need for a procedure to anticipate and pinpoint key problem areas. The framework is structured to contain key elements of a systems project, areas where problems usually occur and which are important to eventual project success. By routinely monitoring these areas, managers can identify problems more quickly and possibly earlier than would be the case without a framework. Use of the factors may also promote easier recognition of problems. In addition, problems which are new, complex, or multi-faceted, may be more easily decomposed into recognizable, manageable components by analyzing them with the framework factors.

IV.2 Function #2 -- Strategy Development

The second major function for the framework is strategy development. The performance of a strategy can be monitored through the elements of the framework. Problems and progress identified in these areas may highlight the need to change some or all of a strategy in order to continue progress towards objectives. If strategy problems occur, the factor structure and the information base available from regular monitoring may then also assist in formulating the changes needed. In addition, the framework may promote better strategy by offering a means to consider longer term perspectives. First, retrospective analysis of the factors may uncover problems that are unidentified by constant attention to immediate circumstances. Similarly, by focusing on specific framework factors during strategy revision, managers may be encouraged to predict the effects of strategy changes in the future. Besides bringing a longer term perspective to problem solving, this can provide an opportunity to identify the conditions or points in time when a strategy should again be reviewed. The longer term cause and effect linkages which are identified through these analyses may also help managers avoid repeating mistakes.

IV.3 Function #3 -- Research

The third function of the framework is to provide a means for researchers to study and contribute to on-going system developments. This reflects the finding that systems researchers see a need for closer contact with actual projects -- partly to gather empirical evidence not easily available through other means and partly to transfer the guidance available in the literature to practitioners who some claim have difficulty using the literature. While these two problems are to some extent exacerbated by the few incentives practitioners have to study their own experiences, it does appear that an approach which unites researchers and practitioners may be a good way to gain the access needed by the research community.

The monitoring framework provides a research opportunity by establishing a data collection process that researchers can use to define and implement studies. By adapting their studies to the framework structure and procedures, researchers can acquire a series of data points at the same time as project managers. They can then use this data to study changes and linkages between factors of interest and to produce valuable insights of use to others.

Researchers may find it beneficial to design their studies with the project managers. Besides helping to ensure continuing access to the project, this approach may provide researchers an opportunity to help managers with problem solving. In addition, researchers might be able to bring in relevant guidance from the literature to help managers. This would help counter the claim that research in the systems literature can not be used.

IV.4 Function #4 -- Documentation

The fourth function of the framework is to document project activities for the administrative and research purposes discussed above. Documentation here means establishing an organized and stable written record of project activities as they occur in key selected areas of the framework. In the short term, this function can help project

managers identify current problems and progress and assist in finding ways to improve the project. For researchers, documentation of project events as they occur is one essential feature of the monitoring approach that makes it attractive over retrospective studies.

Documentation is seen as particularly useful, however, for tracing and analyzing selected factors over long periods of time. For managers, documentation of project activities can be inefficient for short term problem solving. In addition, documentation over the long term can provide several benefits that may be difficult to obtain from short term analyses of immediate actions:

- Routine documentation over the long term can help managers identify special, evolving, or recurring problems.
- Longer term analyses of management actions may help identify successful and unsuccessful approaches, promoting improved strategies or new ideas for current actions.
- Long term documentation can support decisionmaking in the project, particularly for major decisions at the end of phases. These decisions may rely on credible evidence of performance -- unattainable from quick retrospective analyses at the time of the decision.
- Documentation can be used to inform new staff of project history and thus ease the problem of turnover.

For researchers, the long term stream of data available from documentation provides the data base on which to conduct research. In particular, long range documentation provides:

- The evidence upon which to identify and test relationships between important factors in development.
- A basis of developing models of the dynamic, evolving process of a large scale system development presently missing in the literature.
- Empirical evidence about all project stages that is also needed in the literature according to some researchers. This may be especially important for other researchers who need better access to actual projects.

IV.5 Function #5 -- Dissemination

The fifth and final function of the monitoring framework is to

promote dissemination of information about the system development to groups not directly involved in oversight roles. The framework structure and resulting data base can facilitate the transfer of information to these groups by making it easier and quicker for managers to generate the information needed. This of course must be tempered with the need to avoid release of interim or uncertain information which might harm the project.

Managers may find the dissemination function useful both inside and outside a project. For insiders, dissemination of monitoring information can facilitate a common awareness of problems and progress and help promote united actions. In a large project, this may be especially important in coordinating groups working on different parts of a system. Dissemination of pertinent information about progress may also have the secondary effect of promoting or facilitating the monitoring activity. Project staff may come to rely on periodic reports of activities for guiding their own work.

For outsiders, dissemination of monitoring information can be essential to keeping groups informed about progress and building awareness and support for the system. This may be especially critical to user groups not directly involved in the development, but having some indirect role in future system operations. Other important groups can be top level managers or outside institutions who supply resources to the project, periodically review progress, and approve continuations of the work. These people may need current and retrospective reports on development activities, both easily supplied if monitoring has been routinized.

Dissemination is also important to researchers. By providing the opportunity to closely study on-going development activities, the framework can facilitate the transfer of information to the systems literature. For example, the framework can help provide a more dynamic, evolving view of development and other empirical evidence that some researchers believe are needed in this area. Besides helping other researchers, real-time analysis of project activities may offer insights to other system developers who are looking for guidance of use in their own situations.

V. A Model for Implementing the Monitoring Process

V.1 Introduction

The framework has been designed with a set of procedures that managers can use to apply it to specific developments. This process is modeled in Figure V.1 by a flow diagram of events which are generally expected to be a part of monitoring. The model is illustrative; modifications will likely be needed in order to match the process to the project.

V.2 Implementation Procedures

As shown in Figure V.1, the monitoring process begins when project managers identify the need for monitoring and proceed to specify the indicators, procedures, and staff needed to operate the monitoring system. To specify the monitoring indicators, managers need to examine the individual elements of the framework and decide what indicators are needed. For example, under the category of user commitment, managers need to identify events that will indicate use and support for the system. For the system champion factor listed under support, this will involve identifying where champions would arise and what actions will be considered supportive. This identification process will of course be influenced by what managers and others, inside or outside the project, need to know during a development. In addition, managers may find gaps in the elements or alternative factors which better match their specific situation.[1] Analysis of each framework element in this manner will specify where monitoring should occur and what needs to be monitored.

Next, managers need to specify the procedures for monitoring. Some of the considerations include they following:

- How monitoring will be coordinated with major decision points in the project.

---

[1] The framework is intended to be comprehensive. However, this is still the first attempt at defining such a framework. Revisions and additions are expected.

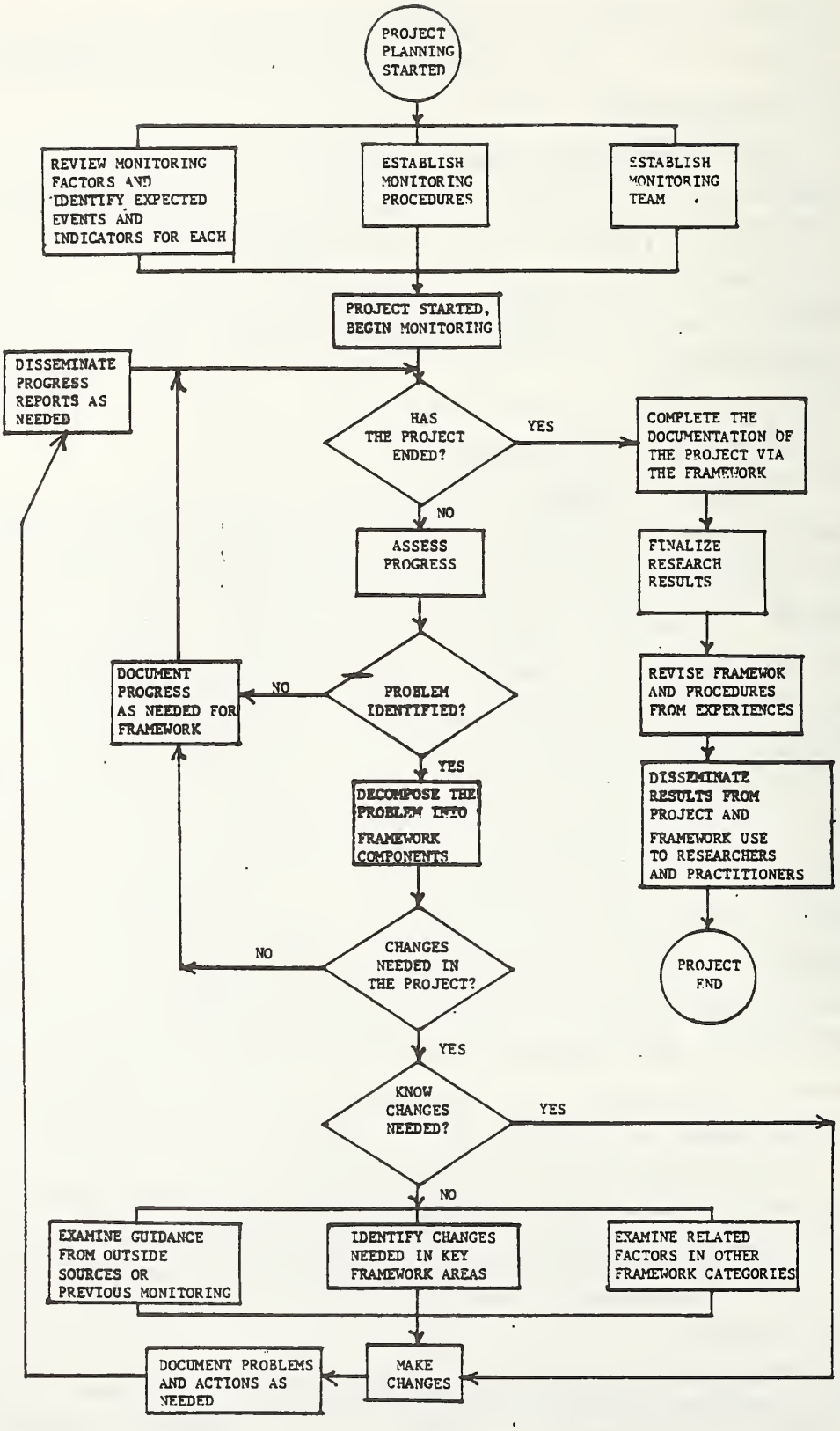
PROJECT PLANNING STARTED
REVIEW MONITORING FACTORS AND IDENTIFY EXPECTED EVENTS AND INDICATORS FOR EACH
ESTABLISH MONITORING PROCEDURES
ESTABLISH MONITORING TEAM
PROJECT STARTED, BEGIN MONITORING
DISSEMINATE PROGRESS REPORTS AS NEEDED
HAS THE PROJECT ENDED?
YES
COMPLETE THE DOCUMENTATION OF THE PROJECT VIA THE FRAMEWORK
NO
ASSESS PROGRESS
FINALIZE RESEARCH RESULTS
DOCUMENT PROGRESS AS NEEDED FOR FRAMEWORK
NO
PROBLEM IDENTIFIED?
REVISE FRAMEWOK AND PROCEDURES FROM EXPERIENCES
YES
DECOMPOSE THE PROBLEM INTO FRAMEWORK COMPONENTS
DISSEMINATE RESULTS FROM PROJECT AND FRAMEWORK USE TO RESEARCHERS AND PRACTITIONERS
NO
CHANGES NEEDED IN THE PROJECT?
PROJECT END
YES
KNOW CHANGES NEEDED?
YES
NO
EXAMINE GUIDANCE FROM OUTSIDE SOURCES OR PREVIOUS MONITORING
IDENTIFY CHANGES NEEDED IN KEY FRAMEWORK AREAS
EXAMINE RELATED FACTORS IN OTHER FRAMEWORK CATEGORIES
DOCUMENT PROBLEMS AND ACTIONS AS NEEDED
MAKE CHANGES

FIGURE V.1 A GENERAL MODEL FOR APPLYING THE MONITORING FRAMEWORK

- How monitoring data will be collected.
- Who will receive monitoring information and what information will be supplied.
- The storage and location of monitoring information.
- What kind of documentation is desired.
- How research activities will be tied to management activities.

Operational details like these areas will establish how monitoring is to be performed and integrated into project management.

Finally, at the same time as these procedures are being explored, managers should also identify the team which will perform the monitoring. While top level project managers are the main users of monitoring information, their role in collecting the information should be minimal. This work should be delegated to the staff supporting project leaders. Managers will have to decide who will be on the team and how the different groups in a development should be represented. It will be beneficial if the team consists of members from all participating groups: the user organization, the development team, and the group of researchers (if present) studying the development. This should help ensure collection and use of monitoring information, while improving the credibility of the monitoring by bringing different perspectives to the process.

After these initial preparations are completed, the monitoring process is started. Problems and progress in specific areas of the framework are identified as they occur or at selected periodic reviews.[2] If the situation is new or not well understood, the monitoring team may need to decompose project events into smaller elements by identifying the relevant factors in each framework category. The relationships between different factors should also be considered. As appropriate, the monitoring team then produces a problem statement for project managers. This procedure is illustrated in Figure V.2.

---

[2] It is assumed here that situations occur which require intervention by project managers.

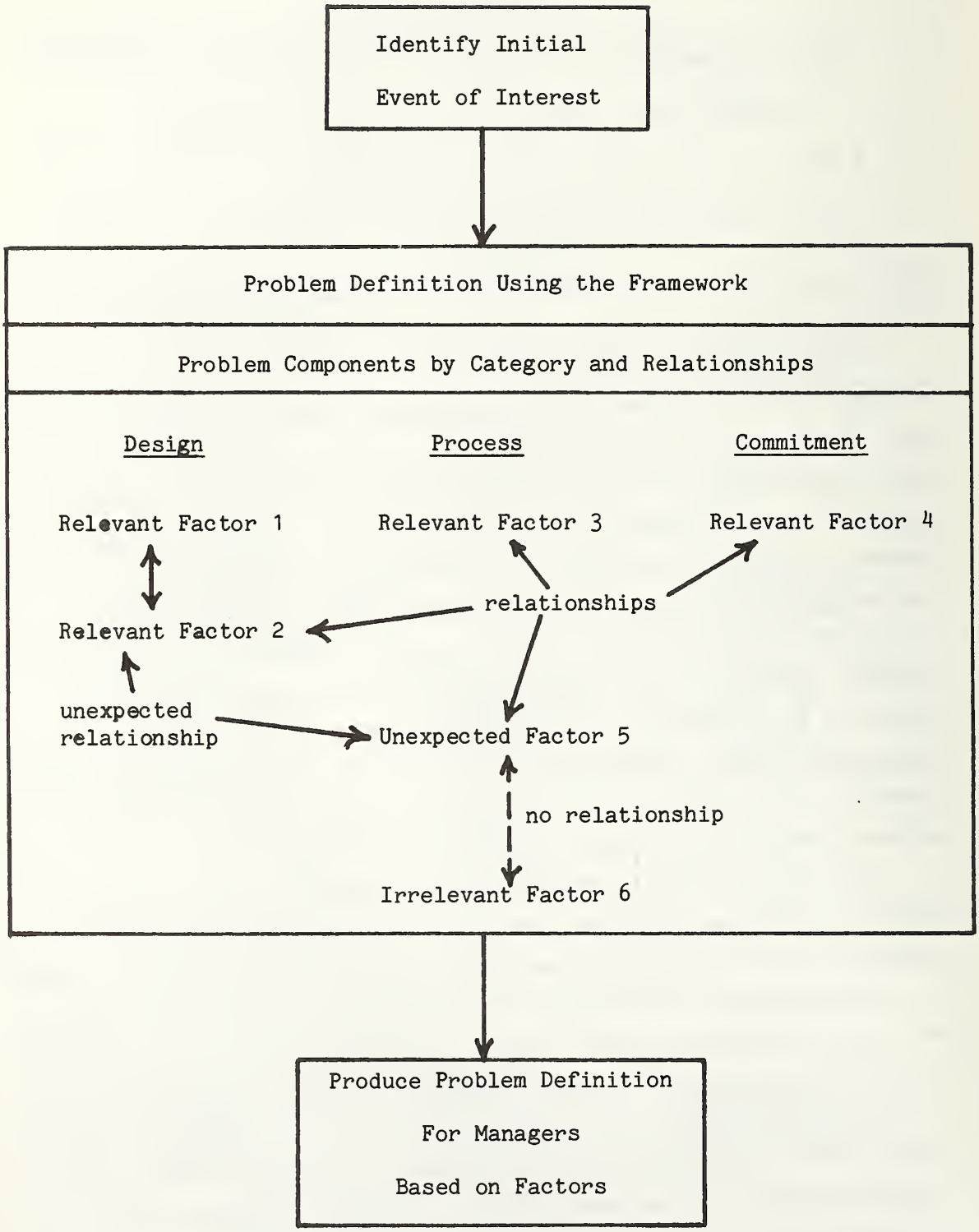

FIGURE V.2 PROBLEM SOLVING USING THE FRAMEWORK

With the problem identified, managers then decide what, if anything, should be done. This is when the strategy development function of the framework can be useful. If managers know what should be changed, then no further analysis is needed and the changes should be made. In this case, the only remaining tasks for the monitoring team are to: document the problems and actions taken for future reference, identify future conditions when these actions should again be reviewed, and disseminate information as needed about the management activity.

However, if the appropriate action is uncertain, further analysis of the problem using the framework may be beneficial for identifying options. In this case, the framework factors may provide a focus both for the problem and the management intervention. Managers can:

- Identify the key components of the problem using the elements of the framework and what actions might be targeted at these areas.
- Identify and examine the relationship between problem elements, using the different framework categories to decompose a complex situation.
- Examine guidance on the problem elements from previous monitoring information, project research underway, or other outside sources.

Once the interventions are specifically identified, the monitoring team then determines how to trace the effects in the future and whether special attention is needed to identify when a review of the changes may be needed (see illustration in Figure V.3). Over the long run, tying the state of the project to management interventions should help the monitoring team identify successful approaches and promote better management control.

After the changes are made and plans for future monitoring are established, the monitoring team documents the activities and disseminates information about them as needed to others inside and outside the project.

This pattern of monitoring continues during the project until monitoring is no longer needed by managers. At this time, the monitor-

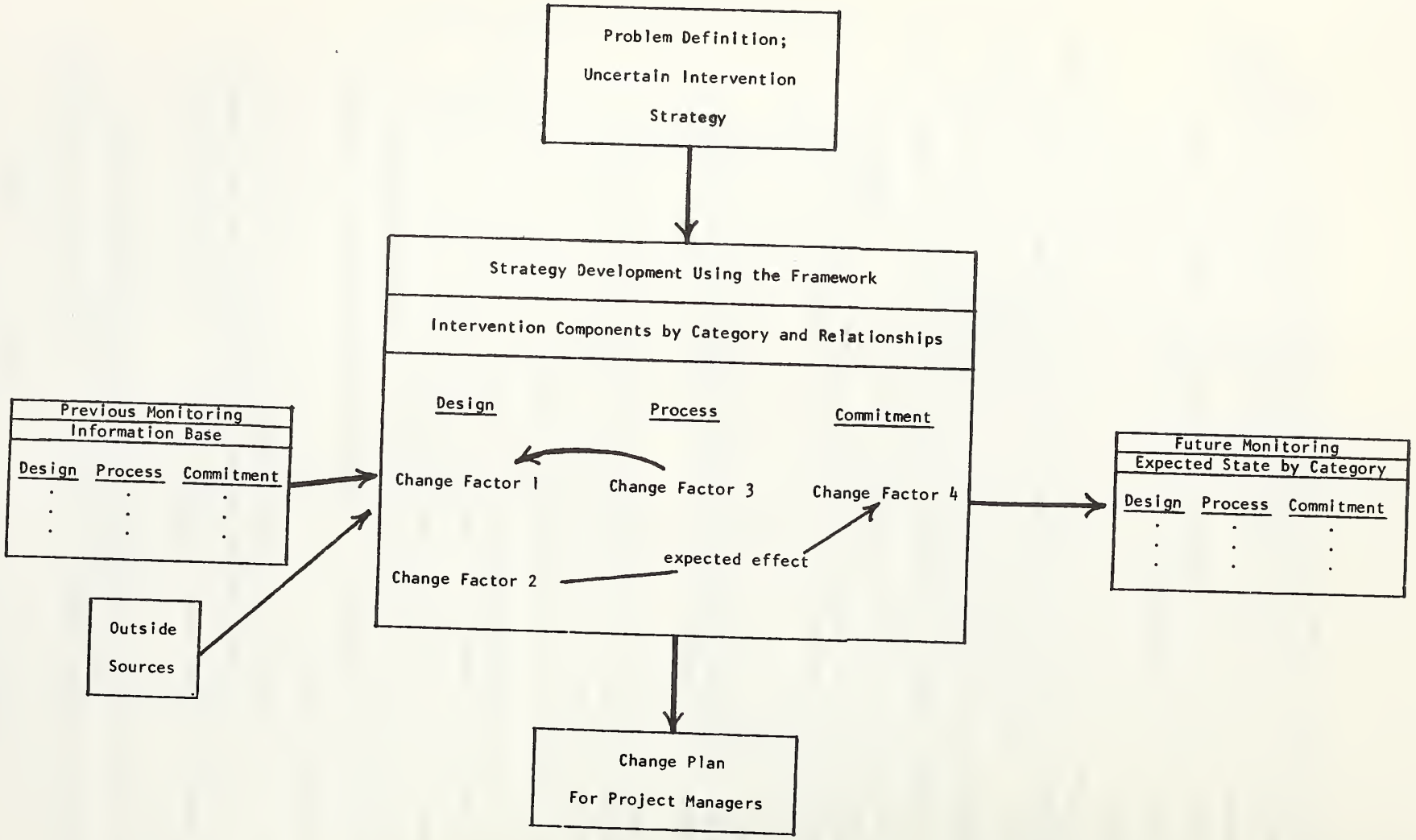

FIGURE V.3 STRATEGY FORMULATION USING THE FRAMEWORK

ing team completes documentation of the project, emphasizing the final state of the project in terms of framework elements. Researchers, if present, then conduct analyses of the data collected by the monitoring team according to their earlier plans. The monitoring team should also review the framework to determine how well it worked and whether changes are needed for future applications.

Finally, the results of the project are disseminated as appropriate to other interested researchers and practitioners.

VI. Conclusion

This manual has presented a process managers and researchers can use to monitor important areas of complex system developments. The manual has described a framework of thirty monitoring factors, five functions monitoring can serve, and a general model for implementing the process.

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 81-2328	2. Performing Organ. Report No.	3. Publication Date September 1981

4. TITLE AND SUBTITLE

A Manual for Designing and Implementing A Process to Monitor Complex System Developments

5. AUTHOR(S)

Stephen D. Garrity

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

See also the related study: Garrity, S.D. Monitoring system development: A framework and application. Washington, D.C.: Experimental Technology Incentives Program, National Bureau of Standards, September 1981. (NBSIR 81-2327)

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

Projects aimed at developing new systems where there are substantial uncertainties as to system requirements, development processes, and ultimate ownership can present project managers with a range of complex, unstructured problems. A process to help identify and solve these problems in a timely, controlled manner is of central importance to the successful conduct of a development with these circumstances.

This document presents a manual for designing and implementing a process to monitor key areas of a complex system development. The manual is derived from an internal study of system developments in the Experimental Technology Incentives Program (ETIP) of the National Burea of Standards.

The proposed monitoring process consists of a framework of thirty factors and a set of five functions which monitoring can serve. The factors are divided into three main categories -- design, process, and user commitment characteristics -- which reflect the general types of problems found to be important in the ETIP environment and the system literature. The functions of the framework, designed to serve both administrative and research purposes, include problem identification, strategy development, research, documentation, and dissemination. A model of how the framework might be implemented within a project is also described.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Administrative experiment; evaluation systems; Experimental Technology Incentives Program; information systems; project management; system acquisition; system development

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

63

15. Price

$8.00